IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Sandon, *et al.*

Examiner: Johnson, Brian P.

Appl. No.: 10/715,688

Group Art Unit: 2183

Filed: 11/18/2003

Docket No.: **END920030075US1**

Title: **TWO DIMENSIONAL ADDRESSING OF A MATRIX-VECTOR REGISTER ARRAY**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

REQUEST FOR RECONSIDERATION

Sir:

This Request for Reconsideration is in response to the Office Action mailed March 14, 2006.

In the Specification:

Please change the title to the following title:

Processor and Method For Processing Matrix Data

In the Claims:

Please cancel claims 15 and 25. Please amend claims 1, 5-7, 9, 13-14, 18-19, and 26-30. Please add new claims 31-32. The claims are as follows.

1. (Currently amended) A processor, comprising M independent vector register files, said M vector register files adapted to collectively store a matrix of L data elements, each data element having B binary bits, said matrix having N rows and M columns, said L=N*M, each column having K subcolumns, said N≥2, said M≥2, said K≥[[1]]2 , said N=K*M, said B≥1, each row of said N rows being addressable, each subcolumn of said K subcolumns being addressable, said processor not adapted to duplicatively store said L data elements.

2. (Original) The processor of claim 1, wherein the processor further comprises M address registers, wherein each address register of the M address registers is associated with a corresponding one of the M vector register files, wherein each of the M vector register files includes an array of N registers, wherein each of the N*M registers of the M vector register files are adapted to store a data element of the L data elements, and wherein each vector register file is independently addressable through its associated address register being adapted to point to one of the N registers of said vector register file.

3. (Original) The processor of claim 2, wherein the data elements of each subcolumn are adapted to be stored in different vector register files, and wherein the data elements of each row are adapted to be stored in different vector register files.

4. (Original) The processor of claim 3, wherein the data elements of each subcolumn are adapted to be stored in different relative register locations of the different vector register files, and wherein the data elements of each row are adapted to be stored in a same relative register location of the different vector register files.

5. (Currently amended) The processor of claim 3, wherein the processor further comprises M multiplexors respectively coupled to the M vector register files, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits, and wherein if the matrix is stored in the M vector register files then:

the M multiplexors are adapted to respond to a command to read a row of the matrix by mapping the data elements of the row from the M vector register files to the row of the matrix in accordance with a read-row mapping algorithm; and

the M multiplexors are adapted to respond to a command to read a subcolumn of the matrix by reading the data elements of the subcolumn from the M vector register files to the subcolumn of the matrix in accordance with a read-subcolumn mapping algorithm.

6. (Currently amended) The processor of claim 3,

wherein the processor further comprises M multiplexors respectively coupled to the M vector register files;

wherein each multiplexor of the M multiplexors comprises a set of binary switches

subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits;

wherein the M multiplexors are adapted to respond to a command to write a row of the matrix by mapping the data elements of the row to the M vector register files in accordance with a write-row mapping algorithm; and

wherein the M multiplexors are adapted to respond to a command to write a subcolumn of the matrix by mapping the data elements of the subcolumn to the M vector register files in accordance with a write-subcolumn mapping algorithm.

7. (Currently amended) The processor of claim 2, wherein the processor further comprises M multiplexors respectively coupled to the M vector register files such that each of the M multiplexors has a different value, and wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits.

8. (Original) The processor of claim 1, wherein the matrix of L data elements are stored in the M vector register files.

9. (Currently amended) A method ~~for processing matrix data~~, comprising:

providing ~~the~~ a processor; ~~and~~

providing M independent vector register files within the processor, said M vector register files collectively storing a matrix of L data elements, each data element having B binary bits, said

matrix having N rows and M columns, said L=N*M, said N=K*M, each column having K subcolumns, said N≥2, said M≥2, said K≥[[1]]2 , said N=K*M, said B≥1, each row of said N rows being addressable, each subcolumn of said K subcolumns being addressable, said processor not duplicatively storing said L data elements; and

respectively coupling M multiplexors coupled to the M vector register files such that each of the M multiplexors has a different value, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits.

10. (Original) The method of claim 9, wherein the method further comprises providing M address registers within the processor, wherein each address register of the M address registers is associated with a corresponding one of the M vector register files, wherein each of the M vector register files includes an array of N registers, wherein each of the N*M registers of the M vector register files stores a data element of the L data elements, and wherein each vector register file is independently addressable through its associated address register being adapted to point to one of the N registers of said vector register file.

11. (Original) The method of claim 10, wherein the data elements of each subcolumn are stored in different vector register files, and wherein the data elements of each row are stored in different vector register files.

12. (Original) The method of claim 11, wherein the data elements of each subcolumn are stored in different relative register locations of the different vector register files, and wherein the data elements of each row are stored in a same relative register location of the different vector register files.

13. (Currently amended) The method of claim 11, wherein the method further comprises providing M multiplexors respectively coupled to the M vector register files, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits, and wherein if the matrix is stored in the M vector register files then:

the M multiplexors are adapted to respond to a command to read a row of the matrix by mapping the data elements of the row from the M vector register files to the row of the matrix in accordance with a read-row mapping algorithm; and

the M multiplexors are adapted to respond to a command to read a subcolumn of the matrix by reading the data elements of the subcolumn from the M vector register files to the subcolumn of the matrix in accordance with a read-subcolumn mapping algorithm.

14. (Currently amended) The method of claim 11,

wherein the method further comprises providing M multiplexors respectively coupled to the M vector register files;

wherein each multiplexor of the M multiplexors comprises a set of binary switches

subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits;

wherein the M multiplexors are adapted to respond to a command to write a row of the matrix by mapping the data elements of the row to the M vector register files in accordance with a write-row mapping algorithm; and

wherein the M multiplexors are adapted to respond to a command to write a subcolumn of the matrix by mapping the data elements of the subcolumn to the M vector register files in accordance with a write-subcolumn mapping algorithm.

15. (Canceled)

16. (Original) The method of claim 9, further comprising addressing a row of the N rows.

17. (Original) The method of claim 9, further comprising addressing a subcolumn of the K*M subcolumns.

18. (Currently amended) A processor, comprising M independent vector register files, said M vector register files adapted to collectively store a matrix of L data elements, each data element having B binary bits, said matrix having N rows and M columns, said L=N*M, each column having K subcolumns, said N≥2, said M≥2, said K≥[[1]]2, said N=K*M, said B≥1, each row of said N rows being addressable, each subcolumn of said K subcolumns being addressable, said matrix including a set of arrays such that each array is a row or subcolumn of the matrix, said

processor adapted to execute an instruction that performs an operation on a first array of the set of arrays, said operation being performed with selectivity with respect to the data elements of the first array.

19. (Currently amended) The processor of claim 18, wherein the processor further comprises M multiplexors respectively coupled to the M vector register files, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits, and wherein the values associated with the M multiplexors control said selectivity.

20. (Original) The processor of claim 18, wherein the processor further comprises M address registers, wherein each address register of the M address registers is associated with a corresponding one of the M vector register files, wherein each of the M vector register files includes an array of N registers, wherein each of the N*M registers of the M vector register files are adapted to store a data element of the L data elements, and wherein each vector register file is independently addressable through its associated address register being adapted to point to one of the N registers of said vector register file.

21. (Original) The processor of claim 18, wherein the instruction is adapted to copy at least one data element of the first array of the set of arrays to a second array of the set of arrays, and wherein the instruction does not insert an exact copy of the first array into the second array.

22. (Original) The processor of claim 18, wherein the instruction is adapted to rearrange the data elements of the first array within the first array.

23. (Original) The processor of claim 18, wherein the processor is not adapted to duplicatively store the L data elements.

24. (Original) The processor of claim 18, wherein the matrix of L data elements are stored in the M vector register files.

25. (Canceled)

26. (Currently amended) ~~The method of claim 25, further comprising~~ A method for processing matrix data, comprising:

providing the processor;

providing M independent vector register files within the processor, said M vector register files collectively storing a matrix of L data elements, each data element having B binary bits, said matrix having N rows and M columns, said L=N*M, each column having K subcolumns, said N≥2, said M≥2, said K≥1, said B≥1, each row of said N rows being addressable, each subcolumn of said K subcolumns being addressable, said matrix including a set of arrays such that each array is a row or subcolumn of the matrix; and

executing an instruction by said processor, said instruction performing an operation on a first array of the set of arrays, said operation being performed with selectivity with respect to the

data elements of the first array; and

providing M multiplexors respectively coupled to the M vector register files, wherein the values associated with the M multiplexors control said selectivity.

27. (Currently amended) The method of claim ~~25~~ 26, wherein the method further comprises providing M address registers within the processor, wherein each address register of the M address registers is associated with a corresponding one of the M vector register files, wherein each of the M vector register files includes an array of N registers, wherein each of the N*M registers of the M vector register files stores a data element of the L data elements, and wherein each vector register file is independently addressable through its associated address register being adapted to point to one of the N registers of said vector register file.

28. (Currently amended) The method of claim ~~25~~ 26, wherein said performing an operation includes copying at least one data element of the first array of the set of arrays to a second array of the set of arrays, and wherein said copying does not insert an exact copy of the first array into the second array.

29. (Currently amended) The method of claim ~~25~~ 26, wherein said performing an operation includes rearranging the data elements of the first array within the first array.

30. (Currently amended) The method of claim ~~25~~ 26, wherein the processor is not duplicatively storing the L data elements.

31. (New) The method of claim 26, said $K \geq 2$, said $N=K*M$.

32. (New) The method of claim 26, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits.

REMARKS

Claim 26 has been rewritten in independent form and is otherwise the same claim 26 that was originally filed..

The Examiner objected to the title, alleging that "a new title is required that is clearly indicative of the invention to which the claims are directed". In response, Applicants have changed the title by amendment herein, such that the new title is clearly indicative of the invention to which the claims are directed.

The Examiner rejected claims 9-17 under 35 U.S.C. § 112, second paragraph.

The Examiner rejected claims 1-20, 23, 27 and 30 under 35 U.S.C. § 102(b) as allegedly being anticipated by Access and Alignment of Data in an Array Processor (herein Lawrie).

The Examiner rejected claims 21-22 and 28-29 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawrie in view of AMD 64-bit Technology (herein AMD).

Applicants respectfully traverse the title objection, § 112 and § 102 rejections with the following arguments.

35 U.S.C. § 112, Second Paragraph

The Examiner rejected claims 9-17 under 35 U.S.C. § 112, second paragraph, as allegedly being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Since claim 15 has been canceled, the rejection of claim 15 under 35 U.S.C. § 112, second paragraph is moot.

The Examiner argues: "Claims 9-17 are method claims, but contain no essential steps for accomplishing a method."

In response, Applicants have amended independent claim 9 such that the recited method comprises active method steps which are within the scope of the preamble.

Based on the preceding arguments, Applicants respectfully maintain that claims 9-14 and 16-17 are not unpatentable under 35 U.S.C. § 112, second paragraph.

35 U.S.C. § 102(b)

The Examiner rejected claims 1-20, 23, 27 and 30 under 35 U.S.C. § 102(b) as allegedly being anticipated by Access and Alignment of Data in an Array Processor (herein Lawrie).

Since claims 15 and 25 have been canceled, the rejection of claims 15 and 25 under 35 U.S.C. § 102(b) is moot.

Claims 1-20 and 23

Applicants respectfully contend that Lawrie does not anticipate claims 1, 9, and 18, because Lawrie does not teach each and every feature of claims 1, 9, and 18.

As a first example of why Lawrie does not anticipate claims 1, 9, and 18, Lawrie does not teach the feature: "said matrix having N rows and M columns, ... each column having K subcolumns, said $N \geq 2$, said $M \geq 2$, said $K \geq 2$, said $N=K*M$". It is clear from the preceding feature of $K \geq 2$ and $N=K*M$ that $N > M$ (i.e., the number of rows N is greater than the number of columns M for the claimed matrix). However, Lawrie's disclosed methodology is specific to a square NxN matrix (see Lawrie, page 100, third line below title for Section A, illustrated examples, discussion throughout) and is non-enabling for a matrix that is not square. Note also, that Lawrie's methodology is formulated to allow access to a diagonal of a matrix and Lawrie's disclosure makes numerous remarks throughout relating to the diagonal of the matrix. However, a diagonal of a matrix is defined only if the matrix is a square matrix. Thus, the constraint in claims 1, 9, and 18 that the matrix is not a square matrix effectively eliminates the use of Lawrie as a reference that can be used to reject claims 1, 9, and 18 under 35 U.S.C. § 102(b).

As a second example of why Lawrie does not anticipate claims 1, 9, and 18, Lawrie does not teach the feature: "each row of said N rows being addressable". The preceding language in claims 1, 8, and 15 requires that each row as a unit be addressable. Lawrie, page 99, section II, second paragraph, lines 4-6 (referred to by the Examiner) merely states that any row can be accessed. However, a row can be accessed by addressing each individual element of the row and does not require addressing the row as a unit. Lawrie does not teach that the row as a unit is addressable. In addition, Lawrie does not teach a mechanism or algorithm for addressing a row as a unit. By not teaching a mechanism or algorithm for addressing a row as a unit, Lawrie's disclosure is non-enabling for addressing a row as a unit.

As a third example of why Lawrie does not anticipate claims 1, 9, and 18, Lawrie does not teach the feature: "each subcolumn of said K subcolumns being addressable". Lawrie's disclosure is specific to whole columns of the matrix to be stored and does not concern itself with individual subcolumns of a column such that each column has at least two subcolumns. Lawrie neither teaches nor provides enablement for the preceding feature of claims 1, 9, and 18.

In addition with respect to claim 9, Lawrie does not the feature: "respectively coupling M multiplexors coupled to the M vector register files such that each of the M multiplexors has a different value, wherein each multiplexor of the M multiplexors comprises a set of binary switches subject to each binary switch being on or off and respectively represented by a binary bit 1 or 0 such that the value of the multiplexor consists of the composite value of said binary bits".

Applicants additionally point out that the Examiner's analysis of claims 1, 9, and 18 is misdirected, because the Examiner's discussion of columns defines a column as being a column of registers. However, a column referred to in claims 1, 9, and 18 claims is a column of the matrix being stored and not a column of registers.

Based on the preceding arguments, Applicants respectfully maintain that Lawrie does not anticipate claims 1, 9, and 18, and that claims 1, 9, and 18 are in condition for allowance. Since claims 2-8 depend from claim 1, Applicants contend that claims 2-8 are likewise in condition for allowance. Since claims 10-14 and 16-17 depend from claim 9, Applicants contend that claims 10-14 and 16-17 are likewise in condition for allowance. Since claims 19-20 and 23 depend from claim 18, Applicants contend that claims 19-20 and 23 are likewise in condition for allowance.

Claims 26 and 28-29

Applicants respectfully contend that Lawrie does not anticipate claim 26, because Lawrie does not teach each and every feature of claim 26.

As a first example of why Lawrie does not anticipate claim 26, Lawrie does not teach the feature: "providing M multiplexors respectively coupled to the M vector register".

The Examiner argues: "Lawrie discloses the processor of claims 18 and 25, wherein the processor further comprises M multiplexors respectively coupled to the M vector register files, and wherein the values associated with the M multiplexors control said selectivity (page 100 col

1 second paragraph).... Note that, according to Wordnet ® 2.0 © 2003 Princeton University, a multiplexor is "a device that can interleave two or more activities"therefore, the mechanism used to choose what values are put into the register file, a mechanism that must exist in this invention, is considered to include a multiplexor for each element.".

In response, Applicants respectfully contend that the Examiner's argument is misdirected, because claim 26 recites that the M multiplexors are respectively coupled to the M vector register files, whereas the Examiner is arguing that the M multiplexors are respectively coupled to individual elements in a single register file. Applicants assert that Lawrie does not teach M multiplexors respectively coupled to the M vector register files.

As a second example of why Lawrie does not anticipate claim 26, Lawrie does not teach the feature: "executing an instruction by said processor, said instruction performing an operation on a first array of the set of arrays, said operation being performed with selectivity with respect to the data elements of the first array ..., wherein the values associated with the M multiplexors control said selectivity".

The Examiner argues: "said processor adapted to execute an instruction that performs an operation on a first array of the set of arrays, said operation being performed with selectivity with respect to the data elements of the first array (page 99 introduction paragraph 1)."

In response, Applicants acknowledge that Lawrie, page 99, introduction, paragraph 1 teaches performing an operation with selectivity with respect to the data elements of the first array, as stated by the Examiner. However, Lawrie does not teach that said selectivity is controlled by the values associated with the M multiplexors.

As a third example of why Lawrie does not anticipate claim 26, Lawrie does not teach the feature: "each row of said N rows being addressable". The preceding language in claims 1, 8, and 15 requires that each row as a unit be addressable. Lawrie, page 99, section II, second paragraph, lines 4-6 (referred to by the Examiner) merely states that any row can be accessed. However, a row can be accessed by addressing each individual element of the row and does not require addressing the row as a unit. Lawrie does not teach that the row as a unit is addressable. In addition, Lawrie does not teach a mechanism or algorithm for addressing a row as a unit. By not teaching a mechanism or algorithm for addressing a row as a unit, Lawrie's disclosure is non-enabling for addressing a row as a unit."

As a fourth example of why Lawrie does not anticipate claim 26, Lawrie does not teach the feature: "each subcolumn of said K subcolumns being addressable". Lawrie's disclosure is specific to whole columns of the matrix to be stored and does not concern itself with individual subcolumns of a column such that each column has at least two subcolumns.

Based on the preceding arguments, Applicants respectfully maintain that Lawrie does not anticipate claim 26, and that claim 26 is in condition for allowance. Since claims 27 and 30 depend from claim 26, Applicants contend that claims 27 and 30 are likewise in condition for allowance.

35 U.S.C. § 103(a)

The Examiner rejected claims 21-22 and 28-29 under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lawrie in view of AMD 64-bit Technology (herein AMD).

Since claims 21-22 depend from claim 18, which Applicants have argued *supra* to not be unpatentable over Lawrie under 35 U.S.C. §102(b), Applicants maintain that claims 21-22 are likewise not unpatentable over Lawrie in view of AMD under 35 U.S.C. §103(a).

Since claims 28-29 depend from claim 26, which Applicants have argued *supra* to not be unpatentable over Lawrie under 35 U.S.C. §102(b), Applicants maintain that claims 28-29 are likewise not unpatentable over Lawrie in view of AMD under 35 U.S.C. §103(a).

CONCLUSION

Based on the preceding arguments, Applicants respectfully believe that all pending claims and the entire application meet the acceptance criteria for allowance and therefore request favorable action. If the Examiner believes that anything further would be helpful to place the application in better condition for allowance, Applicants invites the Examiner to contact Applicants' representative at the telephone number listed below. The Director is hereby authorized to charge and/or credit Deposit Account 09-0457.

Date: 06/14/2006

Jack P. Friedman
Registration No. 44,688

Schmeiser, Olsen & Watts
22 Century Hill Drive - Suite 302
Latham, New York 12110
(518) 220-1850